



## 32. Bundeswettbewerb Informatik 2013/2014

# Die Aufgaben der zweiten Runde

## Allgemeine Hinweise

Herzlichen Glückwunsch zum Erreichen der zweiten Runde! Hier sind die Aufgaben. Sie sind anspruchsvoll, und ihre Bearbeitung ist aufwändig. Aber die Mühe lohnt sich, denn durch Teilnahme an der zweiten Runde

- wirst du sicher sehr viel lernen;
- kannst du dich für die Endrunde qualifizieren;
- kannst du mit einer guten Leistung einen Buchpreis des Verlags O'Reilly gewinnen;
- hast du am Ende eine Arbeit fertig gestellt, die du als so genannte „Besondere Lernleistung“ in die Abiturwertung einbringen kannst;
- kannst du dich (als jüngerer Teilnehmer) um die Teilnahme an einer Deutschen Schülerakademie bewerben;
- hast du die Chance auf eine Einladung zu den „Forschungstagen Informatik 2014“ des Max-Planck-Instituts für Informatik in Saarbrücken.

Wir wünschen also viel Spaß und viel Erfolg bei der Bearbeitung!

Es gibt drei Aufgaben. **Eine Einsendung darf Bearbeitungen zu höchstens zwei Aufgaben enthalten**, deren Bewertung dann das Gesamtergebnis ausmacht. Sollte eine Einsendung Bearbeitungen zu allen drei Aufgaben enthalten, werden wir zwei davon zufällig auswählen und nur diese bewerten.

An dieser Runde dürfen nur Einzelpersonen teilnehmen, die in der ersten Runde in drei Aufgaben insgesamt mindestens 12 Punkte erreicht oder einem Team angehört haben, dem dieses gelungen ist. Gruppenarbeit ist in der zweiten Runde nicht zulässig.

**Einsendeschluss ist der 28. April 2014.**

## Bearbeitung

Die Bearbeitung einer Aufgabe sollte zunächst eine nachvollziehbare und vollständige Lösung aller Teilaufgaben enthalten. **Pluspunkte** für eine höhere Bewertung kannst du erreichen, wenn du die Aufgabe dort, wo es möglich und sinnvoll ist, eigenständig weiterentwickelst. Sinnvoll sind inhaltliche Erweiterungen und Verbesserungen, etwa von Datenstrukturen und Algorithmen; uninteressant sind aufwändige Tricks, z. B. zur reinen Verschönerung

der Benutzungsoberfläche. Begründe für jede Erweiterung, weshalb sie sinnvoll ist und ihre Realisierung eine eigene Schwierigkeit darstellt.

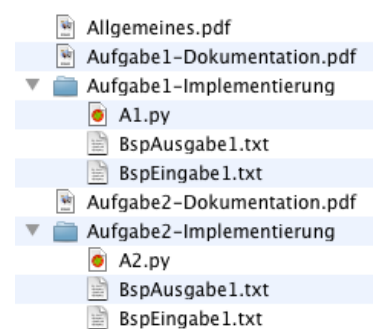
Grundsätzlich gelten die Vorgaben der 1. Runde weiter. Wesentliches Ergebnis der Aufgabebearbeitung ist also eine **Dokumentation**, in der du den *Lösungsweg* sowie die *Umsetzung* des Lösungswegs in das dazugehörige Programm beschreibst. Die Beschreibung des Lösungswegs kann mit Hilfe (halb-)formaler Notationen präzisiert werden, die der Umsetzung mit Verweisen auf die entsprechenden Quellcode-Elemente. In die Dokumentation gehören auch *Beispiele* (Programmein- und -ausgaben oder Zwischenschritte) die zeigen, wie das Programm sich in unterschiedlichen Situationen verhält. Komplettiert wird die Dokumentation durch den *Quelltext*, wobei unwichtige und automatisch generierte Teile weggelassen werden sollten.

Das zweite Ergebnis der Bearbeitung ist die **Implementierung**. Sie besteht aus dem zur Lösung der Aufgabe geschriebenen lauffähigen *Programm* und dem kompletten *Quelltext*. Außerdem können Beispielein- und ausgaben oder weiteres hilfreiches Material der Implementierung beigelegt werden.

Die komplette Dokumentation zu einer Aufgabe muss als ein Dokument eingereicht werden. Dieses Dokument wird für die Bewertung ausgedruckt. **Es kann sein, dass für die Bewertung nur die Dokumentation herangezogen wird!** Diese sollte also einen lückenlosen und nachvollziehbaren Nachweis des Leistungsumfags und der Funktionstüchtigkeit der Programme geben. Ihr Umfang soll sich in Grenzen halten; eine gute Dokumentation vermittelt kurz und präzise alles Nötige, insbesondere die wesentlichen Ideen beim Lösungsweg. Nötig ist alles, was Interessierte mit guten Informatikkenntnissen, die die Aufgabenstellung kennen, wissen müssen, um den Lösungsweg zu verstehen und seine Realisierung nachzuvollziehen. Entscheidend für eine gute Bewertung sind zwar richtige (und sauber umgesetzte) Ansätze, aber die Dokumentation hat schon oft den Ausschlag für oder gegen das Weiterkommen gegeben.

## Einsendung

Die Einsendung erfolgt wieder über das BWINF-PMS ([pms.bwinf.de](http://pms.bwinf.de)). Hochladen kannst du ein max. 10MB großes ZIP-Archiv (z.B. VornameNachname.zip); seine Inhalte sollten so strukturiert sein wie im Bild rechts. Die Dokumentationen müssen als PDF-Dokumente enthalten sein. Dateien in anderen Formaten werden möglicherweise ignoriert. Die Schriftgröße einer Dokumentation muss mindestens 10 Punkt sein, bei Quelltext mindestens 8 Punkt. Auf jeder Seite einer Dokumentation sollen in der Kopfzeile Verwaltungsnummer, Vorname, Name und Seitennummer stehen. Die Verwaltungsnummer steht auf der Teilnahmebescheinigung der ersten Runde.



## Weitere Hinweise

Bei der Bewertung können Programme unter Windows (Vista / 7), Linux und Mac OS X (10.9) ausgeführt werden.

**Fragen zu den Aufgaben** können per E-Mail an [bwinf@bwinf.de](mailto:bwinf@bwinf.de) oder telefonisch unter 0228-378646 (zu üblichen Arbeitszeiten) gestellt werden. Die Antwort auf E-Mail-Anfragen kann sich leicht verzögern. Informationen zur 2. Runde finden sich auf unseren Webseiten ([www.bundeswettbewerb-informatik.de](http://www.bundeswettbewerb-informatik.de)). Auf [einstieg-informatik.de](http://einstieg-informatik.de) werden in der Community sicher wieder viele Teilnehmer über die Aufgaben diskutieren – ohne Lösungsideen auszutauschen.

Allen Teilnehmern der zweiten Runde wird bis Mitte Juni 2014 die Bewertung mitgeteilt. Die Besten werden zur Endrunde eingeladen, die im Frühherbst 2014 an und von der Leuphana Universität Lüneburg ausgerichtet wird. Dort werden die Bundessieger und Preisträger ermittelt und ausgezeichnet. Bundessieger werden in der Regel ohne weiteres Auswahlverfahren in die Förderung der Studienstiftung des deutschen Volkes aufgenommen. Außerdem werden Geld- und Sachpreise vergeben. Der Rechtsweg ist ausgeschlossen.

Zum Schluss noch einmal: Viel Spaß und viel Erfolg!

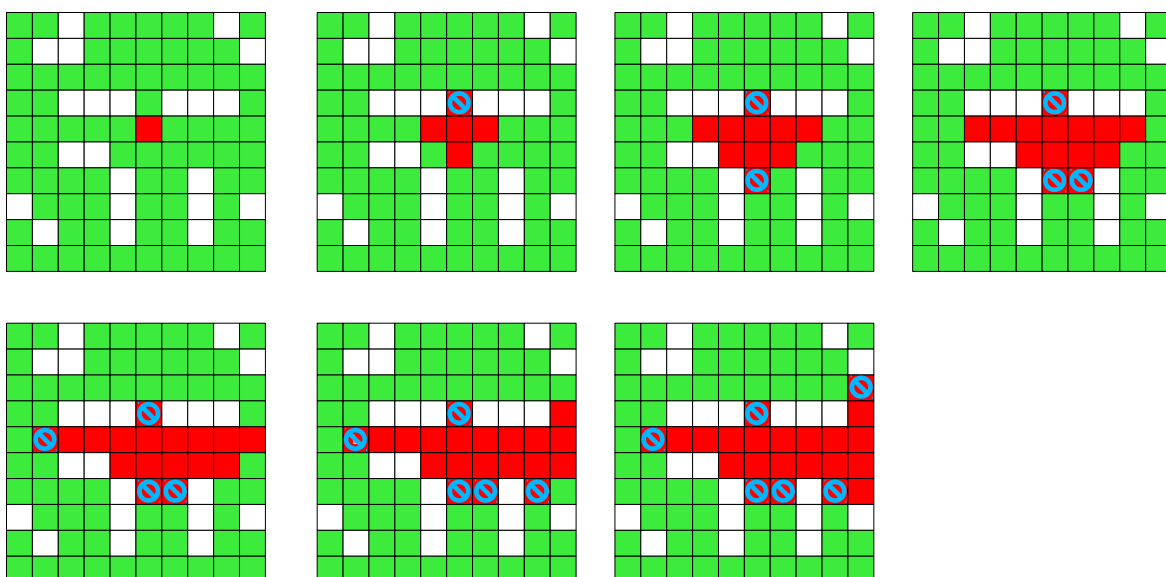
## Aufgabe 1: Buschfeuer

Es brennt in den Pontypandy Mountains, und das Feuer breitet sich rasend schnell aus. Leider sind die Berge nicht per Kraftfahrzeug erreichbar, so dass der Löschhubschrauber der Mountain Rangers verwendet werden muss. Der Hubschrauber kann genug Wasser aufnehmen, um genau ein brennendes Waldstück vollständig zu löschen. Ein gelöscht Waldstück kann nicht mehr brennen.

Deine Aufgabe ist es, den Löscheinsatz des Hubschraubers so zu steuern, dass möglichst wenig Waldstücke in Brand geraten.

Glücklicherweise existieren Brandschneisen im Wald, so dass sich das Feuer nicht beliebig ausbreiten kann. Das gesamte Gebiet kann durch ein Raster modelliert werden, in dem jedes Feld einem Waldstück oder einem Stück einer Brandschneise entspricht. Das Feuer kann sich von einem Feld in einer Stunde auf die vier benachbarten Felder ausbreiten. Für die Aufnahme und den Transport des Wassers benötigt der Hubschrauber ebenfalls eine Stunde.

Die Abbildung zeigt den Verlauf einer Löschkaktion. Weiße Felder stellen Stücke von Brandschneisen dar, rote (dunkle) Felder in Brand geratene Waldstücke. Das Symbol zeigt, welche Waldstücke gelöscht wurden.



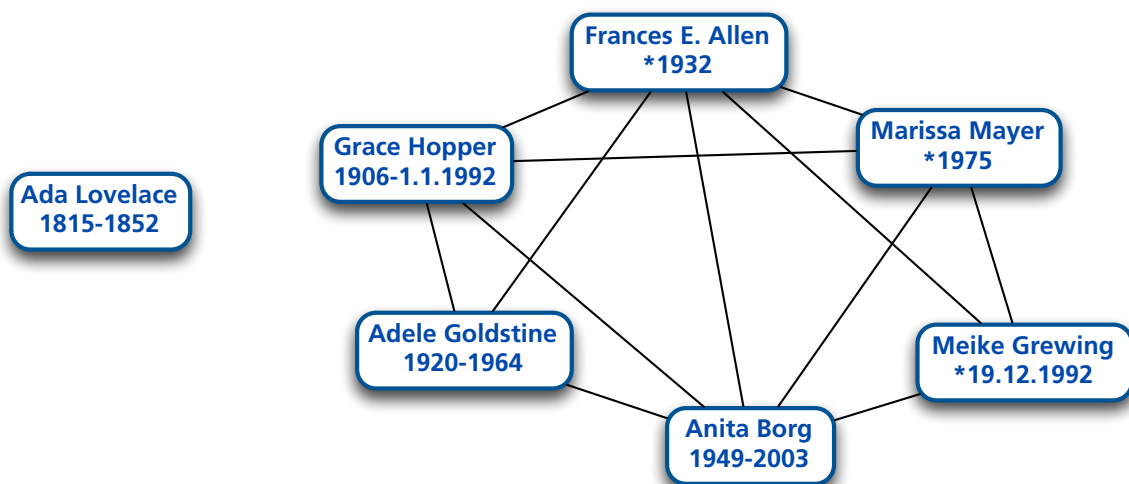
### Aufgabe

Wo muss der Hubschrauber wann Wasser abwerfen, damit möglichst wenige Waldstücke in Brand geraten? Schreibe ein Programm, das diese Frage für gegebene Raster mit Waldstücken, Brandschneisen und einem oder mehreren Brandherden beantworten kann.

Feuerwehrfrau Sheila fragt sich, ob mehr Wald gerettet werden kann, wenn wahlweise auch nicht brennende Waldstücke mit Wasser vor dem Feuer geschützt werden. Was meinst du?

## Aufgabe 2: Lebenslinien

Hobby-Historiker Al Ora – der sich von seinen Freunden gern Herr O-dot nennen lässt – versucht häufig, die Beziehungen innerhalb von Gruppen historischer Persönlichkeiten zu ergründen. Dabei findet er es hilfreich, die Lebensspannen der betrachteten Personen zusammen auf einer Zeitachse einzutragen, um leichter überblicken zu können, wer überhaupt gleichzeitig gelebt hat. Weiter in diese Richtung vordringend definiert er den *Lebensgraphen* einer Gruppe von Personen als den ungerichteten Graphen mit einem Knoten für jede Person, in dem zwei Knoten genau dann durch eine Kante verbunden sind, wenn die Lebensspannen der beiden entsprechenden Personen sich überlappen. Hier sind z.B. die Lebensspannen und der Lebensgraph von Ada Lovelace, Grace Hopper, Adele Goldstine, Frances E. Allen, Anita Borg, Marissa Mayer und Meike Grewing:



Seit einiger Zeit betrachtet Herr O-dot sogar *potenzielle* Lebensgraphen. Ein *PLG* ist ein ungerichteter Graph, der Lebensgraph einer Gruppe von Personen sein könnte, für den es also Zeitspannen gibt, eine für jeden Knoten, so dass zwei Zeitspannen sich genau dann überlappen, wenn es zwischen den beiden entsprechenden Knoten eine Kante gibt.

Nachdem er es länger vermutet hatte, konnte Herr O-dot neulich beweisen, dass es einen ungerichteten Graphen gibt, der kein PLG ist. Allerdings hat er es noch nicht geschafft, einen PLG-Tester zu programmieren, der entscheiden soll, ob ein vorliegender Graph ein PLG ist.

### Aufgabe:

1. Hilf Herrn O-dot, indem du einen möglichst effizienten PLG-Tester entwirfst und implementierst. Wenn ein Eingabegraph ein PLG ist, soll dein Programm nicht nur „Ja“ antworten, sondern auch mögliche „Lebensspannen der Knoten“ ausgeben.
2. Erweitere dein Programm folgendermaßen: Wenn ein Eingabegraph kein PLG ist, soll eine minimale Teilmenge der Knoten ausgegeben werden, so dass bereits diese Knoten mit den Kanten zwischen ihnen keinen PLG bilden.

## Aufgabe 3: Vortänzer Challenge

Aus der Aufgabe „Vortänzer“ der ersten Runde sind sie bekannt: die Roboter des Typs Dancemaster 2000. Endlich ist es für sie soweit, die Dance-Challenge steht an und wird auf dem BWINF-Turnierserver veranstaltet: [turnier.bundeswettbewerb-informatik.de](http://turnier.bundeswettbewerb-informatik.de)

Die Dance-Challenge ist also ein Turnier und besteht aus mehreren Runden. In einer Runde treten jeweils drei Dancemaster 2000 gegeneinander an. Eine Runde besteht aus neun Matches. In einem Match tanzt einer der drei Roboter ein Tanzprogramm vor, und die anderen müssen diesen Tanz imitieren. Der „Vortänzer“ kann für sein Tanzprogramm 10 Zeichen gratis verwenden. Jedes weitere Zeichen kostet 9 Strafpunkte. Die Imitate (also die Tanzprogramme) der beiden „Nachtänzer“ werden nach den Regeln aus Runde 1 mit Strafpunkten belegt. Die beiden Nachtänzer „sehen“ den Tanz des Vortänzers als Folge von Bewegungsbefehlen; Näheres siehe unter „Aufgabe“.

Für eine Runde erhält jeder der drei Roboter eine Startnummer, die zufällig zwischen 1 und 3 gezogen wird. In den neun Matches einer Runde sind die Roboter dann in folgender Reihenfolge Vortänzer: 1 3 2 2 3 1 3 1 2. Die jeweils anderen beiden Roboter sind Nachtänzer.

*Sieger einer Runde* ist der Roboter, der in den neun Matches insgesamt die wenigsten Strafpunkte angehäuft hat. Ein eindeutiger Sieger wird für die Turnierwertung mit 6 Tanztalern (kurz: Tata) belohnt, ein eindeutiger Zweiter noch mit 2 Tata. Gibt es neben dem eindeutigen Sieger zwei Roboter mit Gleichstand, bekommen diese beiden je 1 Tata. Haben zwei Roboter gleich viele Strafpunkte aber weniger als der dritte, bekommen die beiden 4 Tata. Haben alle drei Roboter exakt gleich viele Strafpunkte, bekommt jeder 2 Tata. *Gesamtsieger* der Dance-Challenge ist der Roboter, der im Turnier die meisten Tata ansammelt.

### Aufgabe

Schreibe ein Programm, das die für das Turnier nötigen Funktionen eines Dancemaster 2000 implementiert, und lade es auf den Turnierserver hoch. Das Programm muss aus zwei Teilen bestehen:

**VOR** generiert ein Tanzprogramm für das Vortanzen. Er wird dann aufgerufen, wenn dein Roboter vortanzen soll. Du kannst dich entscheiden, ob VOR immer das gleiche Tanzprogramm generiert (was vermutlich für die anderen Roboter von Vorteil ist) oder ob jeweils ein neues Tanzprogramm generiert wird.

**NACH** bekommt eine vorgetanzte Figur als Folge von Bewegungsbefehlen und muss ein Tanzprogramm erzeugen, mit dem dein Roboter diese Figur möglichst gut imitiert. Im Turniersystem wird die Befehlsfolge als String übergeben, der nur die Zeichen 'F', 'B', 'l', 'r' und '-' enthält. Dieser String kann maximal 255 Zeichen/Befehle lang sein; die Batterie des Dancemaster 2000 hat leider eine begrenzte Kapazität.

*Hinweis:* Voraussetzung für die Bewertung deiner Bearbeitung ist, dass du in der Dokumentation deutlich (am besten zuallererst) den Namen deines für das Turnier angemeldeten Programms (deiner KI) und deinen Username im Turniersystem angibst.